

Iterative proportional scaling revisited: a modern optimization perspective on big count data

Yiyuan She and Shao Tang

Department of Statistics, Florida State University

Abstract

We revisit the classic iterative proportional scaling (IPS) for contingency table analysis, from a modern optimization perspective. In contrast to the criticisms made in the literature, we show that based on a coordinate descent characterization, IPS can be slightly modified to deliver coefficient estimates, and from a majorization-minimization standpoint, IPS can be extended to handle log-affine models with general designs. The optimization techniques help accelerate IPS to provide highly salable algorithms for big count data applications, and can adapt IPS to shrinkage estimation to deal with a large number of variables.

1 Introduction

Count data are ubiquitous in modern statistical applications. Such data are often cross-classified into contingency tables, where iterative proportional scaling (IPS) can be applied as a standard tool ([Fienberg and Meyer, 2006](#)). IPS was firstly used by [Deming and Stephan \(1940\)](#) to adjust a contingency table to obey prescribed column and row margins, the problem of which is nowadays referred to as matrix raking. IPS can be applied to Kullback–Leibler (KL) divergence minimization with linear constraints ([Ireland and Kullback, 1968](#)), and Poisson log-linear model fitting on multi-way tables ([Bishop et al., 1975](#)). There is an interesting duality between these two

types of problems (Good, 1963), although the Lagrange multiplier technique does not apply (Csiszár, 1975). Theoretical studies regarding the convergence properties of IPS undergo a long history and we refer to Pukelsheim (2014) for a comprehensive survey. To name a few, Fienberg (1970) takes a geometric view, assuming that all entries in the input table are positive, while Haberman (1974) and Bishop et al. (1975) are among the first to use the ascent property of the associated Poisson log-likelihood. It is worth mentioning that the problem becomes much more challenging if IPS operates on a table with zero entries (Sinkhorn and Knopp, 1967; Csiszár, 1975).

Although the standard version of IPS is derived for contingency tables, it has some useful extensions. For example, Darroch and Ratcliff (1972) propose the generalized iterative scaling (GIS) which can fit Poisson log-linear models with non-negative designs. Later, Pietra et al. (1997) develop a more relaxed improved iterative scaling (IIS), which does not require all rows of the design matrix must sum to one. The investigation of IPS has led to an abundance of applications in computer science, economics and mathematics (Sinkhorn, 1964; Lahr and De Mesnard, 2004; Rote and Zachariasen, 2007).

Today, IPS becomes somehow outdated and is less frequently used by statisticians (Agresti, 2012). This is largely due to the outstanding performance and generality of Newton-type optimization algorithms. In fact, Newton-Raphson is the default routine in most software for fitting generalized linear models (GLMs), and has quadratic convergence in contrast to the linear convergence rate of IPS. Nevertheless, the traditional IPS algorithm may be motivating and influential in the **Big Data** era. Consider a five-way table of size $10 \times 10 \times 10 \times 10 \times 10$. The associated Poisson model including all up to three-way interactions has 8,146 independent parameters. The corresponding Hessian matrix has more than 10^7 entries, leading to prohibitively high computational cost when using Newton-Raphson. Quasi-Newton methods are more computationally economical, but still fail easily when the model is of high dimensionality. First-order methods (such as gradient descent) are typically more scalable. In our problem, however, there exists no universal stepsize due to the unbounded Hessian. This means that line search methods must be adopted, while evaluating the function value or the gradient could be very expensive on big data.

In comparison with Newton-type and gradient descent methods, IPS has certain benefits in computation. We illustrate the procedure via a toy example to give more intuition. Consider a three-way table m_{ijk} of size $m_1 \times m_2 \times m_3$ with the categorical variables denoted by X_1, X_2, X_3 and assume

a Poisson log-linear model (X_1X_2, X_1X_3, X_2X_3) with margins m_{ij+} , m_{i+k} and m_{+jk} as sufficient statistics (Agresti, 2012). One can call IPS, starting with initial counts $[\mu_{ijk}^{(0)}]_{1 \leq i \leq m_1, 1 \leq j \leq m_2, 1 \leq k \leq m_3}$, to adjust the estimated counts in a multiplicative manner iteratively:

$$\mu_{ijk}^{(3t+1)} = \mu_{ijk}^{(3t)} \frac{m_{ij+}}{\mu_{ij+}^{(3t)}}, \mu_{ijk}^{(3t+2)} = \mu_{ijk}^{(3t+1)} \frac{m_{i+k}}{\mu_{i+k}^{(3t+1)}}, \mu_{ijk}^{(3t+3)} = \mu_{ijk}^{(3t+2)} \frac{m_{+jk}}{\mu_{+jk}^{(3t+2)}}, \quad (1)$$

where $t \geq 0$ is the current iteration index. Clearly, all calculations in (1) are “in-place” (with no need of auxiliary variables) and no line search is required. The whole procedure has memory efficiency, scalability and implementation ease.

IPS is subject to some serious criticisms in the literature, see, e.g., Agresti (2012). Firstly, although estimated expected values of counts are computed, IPS does not deliver any coefficient estimate nor asymptotic covariance estimate. Secondly, this scaling procedure as presented in (1), though elegant, has a somewhat narrow scope and may encounter difficulties in more general scenarios like log-linear models with possibly negative designs (Section 4) or shrinkage estimation (Section 5.2). Thirdly, IPS is cost-effective per iteration but often requires a large number of iterations to converge.

In this work, we study IPS from a modern optimization perspective. We aim to improve and generalize IPS to overcome some aforementioned obstacles. We show that IPS implicitly involves a coefficient-update step and adding it back leads to a novel coordinate decent (CD) characterization of this classical method. This point of view makes it possible to design generalizations and accelerations. We also show that IPS is closely connected to majorization-minimization (MM) algorithms (Hunter and Lange, 2004), which are popular extensions of the EM algorithms among the statistical community. The MM principle successfully extends IPS to general Poisson log-affine models, with the celebrated GIS and IIS taken as two particular instances. Furthermore, these optimization techniques lead to efficient block-wise IPS as well as some shrinkage variants for high dimensional estimation.

Notation. Given $N \in \mathbb{N}$, we define $[N] = \{1, 2, \dots, N\}$. We use bold lower-case and upper-case symbols to denote column vectors and matrices, respectively, i.e. $\mathbf{x} = [x_i] \in \mathbb{R}^N$ and $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{N \times p}$, where $i \in [N]$ and $j \in [p]$. The inner product between two vectors \mathbf{x} and \mathbf{y} is $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$, their Hadamard product is denoted by $\mathbf{x} \circ \mathbf{y}$, and their component-wise division is denoted by $\mathbf{x} \oslash \mathbf{y}$. Given any matrix \mathbf{X} , we denote by x_{i+} the i th row sum $\sum_j x_{ij}$.

For notational ease, we extend functions and operations to vectors component-wisely. For example, given $\mathbf{x} = [x_i] \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$, $\exp(\mathbf{x})$ stands for $[\exp(x_i)]$, and $\mathbf{x} \succeq \mathbf{y}$ indicates that $x_i \geq y_i$ for all $i \in [N]$. The design matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$ is frequently partitioned into columns or rows $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_p] = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N]^T$ with $\mathbf{x}_j \in \mathbb{R}^N$ and $\tilde{\mathbf{x}}_i \in \mathbb{R}^p$. Given two square matrices \mathbf{X} and \mathbf{Y} , $\mathbf{X} \succeq \mathbf{Y}$ means $\mathbf{X} - \mathbf{Y}$ is positive semi-definite. We denote by $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_2$ the ℓ_1 -norm and the ℓ_2 -norm of \mathbf{x} , respectively. The spectral norm and the infinity norm of a matrix \mathbf{X} are defined as $\|\mathbf{X}\|_2 = \sigma_{\max}(\mathbf{X})$ (the largest singular value of \mathbf{X}) and $\|\mathbf{X}\|_\infty = \max_i \sum_j |x_{ij}|$, respectively.

2 Model Setting

Given a contingency table, one can introduce a design matrix to characterize the IPS problem in the framework of Poisson log-linear models. Concretely, let X_k ($1 \leq k \leq r$) be the k th categorical variable taking values in $[m_k]$ and r be the total number of categorical variables. Introduce dummy variables $X_k^{(\ell_k)} \triangleq I(X_k = \ell_k)$ for $\ell_k = 2, 3, \dots, m_k$ and let $\mathbf{X}_k = [X_k^{(2)}, X_k^{(3)}, \dots, X_k^{(m_k)}]$. Then the model matrix containing all main effects has form $[\mathbf{1}, \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_r]$. Similarly, the model matrix including all two-way interactions has the following form $[\mathbf{X}_1 * \mathbf{X}_2, \dots, \mathbf{X}_1 * \mathbf{X}_r, \mathbf{X}_2 * \mathbf{X}_3, \dots, \mathbf{X}_2 * \mathbf{X}_r, \dots, \mathbf{X}_{r-1} * \mathbf{X}_r]$, where $\mathbf{X}_j * \mathbf{X}_k \triangleq [X_j^{(2)} X_k^{(2)}, \dots, X_j^{(2)} X_k^{(m_k)}, \dots, X_j^{(m_j)} X_k^{(m_k)}]$. Higher-order interactions can be included as well.

Given an arbitrary $\mathbf{X} \in \mathbb{R}^{N \times p}$, a Poisson log-linear model with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ can be written as $\boldsymbol{\mu} = \exp(\mathbf{X}\boldsymbol{\beta})$, where $\boldsymbol{\beta}$ denotes the coefficient vector. In some applications, e.g., rate data analysis and matrix raking, an extra offset $\mathbf{q} \succeq \mathbf{0}$ is required to specify a *log-affine* model: $\boldsymbol{\mu} = \mathbf{q} \circ \exp(\mathbf{X}\boldsymbol{\beta})$. We always allow β_j to take $\pm\infty$, i.e., $\boldsymbol{\beta} \in \mathbb{R}^p$ with $\mathbb{R} = [-\infty, \infty]$ (the extended real line).

Let $\mathbf{n} \in \mathbb{R}^N$ denote the observed counts. Then the maximum likelihood (ML) estimation problem of $\boldsymbol{\mu}$ can be formulated according to the log-affine model:

$$\min_{\boldsymbol{\mu}} l(\boldsymbol{\mu}) \triangleq -\langle \mathbf{n}, \log \boldsymbol{\mu} \rangle + \langle \mathbf{1}, \boldsymbol{\mu} \rangle \text{ s.t. } \boldsymbol{\mu} = \mathbf{q} \circ \exp(\boldsymbol{\eta}), \boldsymbol{\eta} \in \bar{\mathcal{R}}_{\mathbf{X}}, \quad (2)$$

where $\bar{\mathcal{R}}_{\mathbf{X}} = \{\sum_{j=1}^p \beta_j \mathbf{x}_j : \beta_j \in \mathbb{R}\}$ denotes the closure of the range of \mathbf{X} and the conventions $\log 0 = -\infty$ and $0 \cdot (\pm\infty) = 0$ are adopted. For convenience, the constraint region is denoted by $\mathcal{M} \triangleq \{\boldsymbol{\mu} \mid \boldsymbol{\mu} = \mathbf{q} \circ \exp(\boldsymbol{\eta}), \boldsymbol{\eta} \in \bar{\mathcal{R}}_{\mathbf{X}}\}$.

As suggested by [Lauritzen \(1996\)](#), theoretically, one needs to consider (2) in a slightly more restricted manner in order to guarantee the convergence of IPS:

$$\min_{\boldsymbol{\mu}} l(\boldsymbol{\mu}) \triangleq -\langle \mathbf{n}, \log \boldsymbol{\mu} \rangle + \langle \mathbf{1}, \boldsymbol{\mu} \rangle \text{ s.t. } \boldsymbol{\mu} \in \bar{\mathcal{M}} \cap \mathcal{M}^*, \quad (3)$$

where $\mathcal{M}^* \triangleq \{\boldsymbol{\mu} \mid l(\boldsymbol{\mu}) < +\infty\}$, meaning that $n_i > 0$ implies $\mu_i > 0$ ($1 \leq i \leq N$).

For simplicity, the following assumptions are made throughout the paper: (i) $\mathbf{n} \neq \mathbf{0}$, (ii) $\mathbf{x}_j \neq \mathbf{0}$, $\forall j \in [p]$, (iii) $\tilde{\mathbf{x}}_i \neq \mathbf{0}$, $\forall i \in [N]$, (iv) $\bar{\mathcal{M}} \cap \mathcal{M}^* \neq \emptyset$. Assumption (i) is trivial. (ii) and (iii) are without loss of generality, since one can drop the corresponding trivial predictors or observations. (iv) ensures finite likelihood for at least one point in $\bar{\mathcal{M}}$, and always holds in real-life applications according to our experience. It is worth mentioning that, under (iv), one can remove certain observations to ensure $\mathbf{q} \succ \mathbf{0}$, without affecting coefficient estimation. In fact, for any $\boldsymbol{\mu} = [\mu_i] \in \bar{\mathcal{M}} \cap \mathcal{M}^*$, $q_i = 0$ implies $\mu_i = 0$ and further $n_i = 0$; so the i th observation does not contribute to the objective function in (2) and can be excluded in optimization. This feature greatly simplifies computation and analysis ([Kurras, 2015](#)).

To facilitate the discussion of extensions of IPS, we specify three types of design matrices:

$$(a) \ x_{ij} = 0 \text{ or } 1 \text{ (binary)}, \ (b) \ x_{ij} \geq 0 \text{ (non-negative)}, \ (c) \ x_{ij} \in \mathbb{R} \text{ (general)}. \quad (4)$$

Clearly, the design matrices derived from contingency table models are special cases of the binary setting.

3 A Coordinate Descent Characterization

3.1 Coefficient-driven IPS

IPS is often criticized for not being able to deliver a coefficient estimate. We will show, however, that this is not true and IPS includes an implicit coefficient-update step.

First, although the objective function of IPS is conventionally formulated in $\boldsymbol{\mu}$ (cf. (2)) or $\log \boldsymbol{\mu}$ ([Haberman, 1974](#)), it is arguably more convenient to rewrite (2) in terms of $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) = -\langle \mathbf{n}, \mathbf{X}\boldsymbol{\beta} \rangle + \langle \mathbf{q}, \exp(\mathbf{X}\boldsymbol{\beta}) \rangle. \quad (5)$$

Algorithm 1 IPS as cyclic coordinate descent.

Input \mathbf{n} , \mathbf{q} and \mathbf{X}

Initialize $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^p$, $t \leftarrow 0$

```

1: while not converged do
2:    $\boldsymbol{\mu}^{(t,0)} \leftarrow \mathbf{q} \circ \exp(\mathbf{X}\boldsymbol{\beta}^{(t)})$ 
3:   for  $j = 1, 2, \dots, p$  do
4:      $\beta_j^{(t+1)} \in \arg \min_{\beta_j} l(\beta_1^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j, \beta_{j+1}^{(t)}, \dots, \beta_p^{(t)})$ 
     Binary case: cf. (9)
5:      $\boldsymbol{\mu}^{(t,j)} \leftarrow \boldsymbol{\mu}^{(t,j-1)} \circ \exp[\mathbf{x}_j(\beta_j^{(t+1)} - \beta_j^{(t)})]$ 
     Binary case: cf. (10)
6:   end for
7:    $\boldsymbol{\beta}^{(t+1)} \leftarrow [\beta_1^{(t+1)}, \dots, \beta_p^{(t+1)}]^T$ ,  $t \leftarrow t + 1$ 
8: end while
9: return  $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(t-1,p)}$ ,  $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$ 

```

The convenience can be partially observed from the initialization condition which requires $\boldsymbol{\mu}^{(0)}$ to take the form of $\mathbf{q} \circ \exp(\boldsymbol{\eta}^{(0)})$ for some $\boldsymbol{\eta}^{(0)}$ in $\bar{\mathcal{R}}_{\mathbf{X}}$ (Fienberg and Meyer, 2006). For this convex optimization problem, we can apply a simple cyclic coordinate descent (CD) algorithm, which updates β_j according to the following formula with the other coordinates fixed

$$\beta_j^{(t+1)} \in \arg \min_{\beta_j} l(\beta_1^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j, \beta_{j+1}^{(t)}, \dots, \beta_p^{(t)}). \quad (6)$$

Define $\boldsymbol{\mu}^{(t,j-1)} \triangleq \mathbf{q} \circ \exp(\mathbf{X}\boldsymbol{\beta}^{(t,j-1)})$ with $\boldsymbol{\beta}^{(t,j-1)} \triangleq [\beta_1^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j^{(t)}, \beta_{j+1}^{(t)}, \dots, \beta_p^{(t)}]^T$. It is easy to show that $\beta_j^{(t+1)}$ satisfies the equation

$$(\boldsymbol{\mu}^{(t,j-1)})^T \{\mathbf{x}_j \circ \exp[\mathbf{x}_j(\beta_j - \beta_j^{(t)})]\} - \mathbf{x}_j^T \mathbf{n} = 0. \quad (7)$$

Algorithm 1 shows the details based on cyclic coordinate update.

The solution to (7) has an explicit form in some special cases. For example, if we assume the design is derived from a contingency table model, or more generally, \mathbf{X} is binary (cf. (4)), then the following relation holds

$$x_{ij} \exp[x_{ij}(\beta_j - \beta_j^{(t)})] = x_{ij} \exp(\beta_j - \beta_j^{(t)}). \quad (8)$$

It follows that $\mathbf{x}_j \circ \exp[(\beta_j - \beta_j^{(t)})\mathbf{x}_j] = \mathbf{x}_j \exp(\beta_j - \beta_j^{(t)})$ and

$$\beta_j^{(t+1)} = \beta_j^{(t)} + \log[\mathbf{x}_j^T \mathbf{n} / (\mathbf{x}_j^T \boldsymbol{\mu}^{(t,j-1)})]. \quad (9)$$

Step 5 in Algorithm 1 then becomes

$$\boldsymbol{\mu}^{(t,j)} = \boldsymbol{\mu}^{(t,j-1)} \circ \exp\{\mathbf{x}_j \log[(\mathbf{x}_j^T \mathbf{n})/(\mathbf{x}_j^T \boldsymbol{\mu}^{(t,j-1)})]\}, \quad (10)$$

or equivalently, $\mu_i^{(t,j)} = (\langle \mathbf{x}_j, \mathbf{n} \rangle / \langle \mathbf{x}_j, \boldsymbol{\mu}^{(t,j-1)} \rangle) \mu_i^{(t,j-1)}$ if $x_{ij} = 1$, and $\mu_i^{(t,j)} = \mu_i^{(t,j-1)}$ otherwise ($1 \leq i \leq N$), which is exactly the IPS algorithm. In the literature, [Haberman \(1974\)](#) shows that IPS is a cyclic ascent method in updating $\log \boldsymbol{\mu}$. To the best of our knowledge, formulating IPS as cyclic CD on $\boldsymbol{\beta}$ is new.

It is clear that Algorithm 1 provides more flexibility in initialization. For instance, $\boldsymbol{\mu}^{(0)} = \mathbf{q}$ is not necessary, starting with an arbitrary $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^p$ suffices. In our experience, a properly chosen initial point can reduce the computational time substantially in large-scale data problems. The algorithm not only offers an easy fix of IPS to deliver $\hat{\boldsymbol{\beta}}$, but also suggests efficient ways to update multiple components of $\boldsymbol{\mu}$ at one time. Specifically, there is no need to adjust all cell values at each epoch, even in sophisticated model settings; when β_j is changed, only the μ_i 's with $x_{ij} \neq 0$ need to be updated.

In history, people conjectured that IPS minimizes Pearson's X^2 -statistic, later known to be incorrect. It is well known that X^2 converges to the asymptotic χ^2 -distribution faster than G^2 . Fortunately, our coordinate descent standpoint gives a simple modification of IPS to solve the problem $\min_{\boldsymbol{\beta}} \sum_i [n_i - \mu_i(\boldsymbol{\beta})]^2 / \mu_i(\boldsymbol{\beta})$ with $\mu_i(\boldsymbol{\beta}) = q_i \exp(\tilde{\mathbf{x}}_i^T \boldsymbol{\beta})$. Similar to the derivation of (7), given the other coordinates, β_j can be updated by solving

$$\sum_i x_{ij} \mu_i^{(t,j-1)} \exp[x_{ij}(\beta_j - \beta_j^{(t)})] - \sum_i \frac{x_{ij} n_i^2}{\mu_i^{(t,j-1)} \exp[x_{ij}(\beta_j - \beta_j^{(t)})]} = 0.$$

It is easy to see that with a binary design, the equation has a closed-form solution and the iterative proportional scaling of $\boldsymbol{\mu}$ is given by

$$\boldsymbol{\mu}^{(t,j)} = \boldsymbol{\mu}^{(t,j-1)} \circ \exp\{0.5 \mathbf{x}_j \log[\mathbf{x}_j^T (\mathbf{n} \circ \mathbf{n} \oslash \boldsymbol{\mu}^{(t,j-1)}) / (\mathbf{x}_j^T \boldsymbol{\mu}^{(t,j-1)})]\}.$$

The differences in comparison with (10) are seen in the additional factor of 0.5, and $\mathbf{n} \circ \mathbf{n} \oslash \boldsymbol{\mu}^{(t,j-1)}$ in place of \mathbf{n} .

3.2 Convergence properties

The coordinate descent characterization facilitates some theoretical studies of convergence properties of IPS. First, we have a natural outcome for Algorithm 1 with a general design \mathbf{X} .

Theorem 1. *For the sequence of iterates $\{\boldsymbol{\beta}^{(t)}\}_{t=0}^{\infty}$ generated by Algorithm 1, the associated function values $l(\boldsymbol{\beta}^{(t)})$ are monotonically non-increasing for $t \geq 0$. In particular, if the first column of \mathbf{X} corresponds to the intercept, then the G^2 -statistic evaluated on $\boldsymbol{\mu}^{(t,1)}$, i.e., $2 \sum_i n_i \log(n_i/\mu_i^{(t,1)})$, is monotonically non-increasing.*

The proof is straightforward and is omitted. Many researchers have obtained similar results on contingency tables (e.g., Bishop et al. (1975); Lauritzen (1996)). We now see that the conclusion directly follows from the coordinate descent nature of the algorithm design, and is not restricted to contingency tables.

Moreover, because of the convexity of the problem, under some regularity conditions, the convergence of the sequence of iterates is readily at hand.

Theorem 2. *Suppose there exists a unique solution $\hat{\boldsymbol{\beta}}$ to problem (5) with finite norm. Then the sequence of iterates $\{\boldsymbol{\beta}^{(t)}\}_{t=0}^{\infty}$ generated by Algorithm 1 has a unique limit point $\hat{\boldsymbol{\beta}}$, and the rate of convergence is at least linear.*

The same rate was first shown by Fienberg (1970) on contingency tables. Our theorem builds upon the studies of decent methods (Luo and Tseng, 1992). In consideration of recent advances (Tseng, 2001; Beck and Tetrushvili, 2013; Razaviyayn et al., 2013; Wright, 2015), one can possibly relax the assumption in certain ways, but we will not pursue further in this work. We refer to Fienberg and Rinaldo (2012) for a systematic study of existence and uniqueness of MLE. In practice, we recommend adding a mild ridge-type penalty (cf. Section 5.2), which ensures the assumption and usually enhances numerical stability.

3.3 Randomization

CD algorithms have recently become popular in the field of statistics, especially in high dimensional applications (Friedman et al., 2007) due to their simplicity and low-complexity operations at each iteration. But they often converge much slower than Newton’s method. Below, we discuss how to update coordinates smartly to reduce the number of iterations.

To ensure non-increasing function values, the cyclic update in Algorithm 1 is not a must; in fact, one can choose the coordinate with the “largest” descent, $j = \arg \max_j |\nabla_j l(\boldsymbol{\beta}^{(t)})|$, the so-called Gauss-Southwell (GS) rule. This is usually the recommended rule and has theoretical support (Nutini

Algorithm 2 A-IPS.

Input n, \mathbf{q} and \mathbf{X} **Initialize** $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^p, t \leftarrow 0$

```
1: while not converged do
2:    $\boldsymbol{\mu}^{(t,0)} \leftarrow \mathbf{q} \circ \exp(\mathbf{X}\boldsymbol{\beta}^{(t)})$ 
3:   for  $j$  in Perm[ $p$ ] do
4:      $\beta_j^{(t+1)} \in \arg \min_{\beta_j} l(\beta_1^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j, \beta_{j+1}^{(t)}, \dots, \beta_p^{(t)})$ 
5:      $\boldsymbol{\mu}^{(t,j)} \leftarrow \boldsymbol{\mu}^{(t,j-1)} \circ \exp[\mathbf{x}_j(\beta_j^{(t+1)} - \beta_j^{(t)})]$ 
6:   end for
7:    $\boldsymbol{\beta}^{(t+1)} \leftarrow [\beta_1^{(t+1)}, \dots, \beta_p^{(t+1)}]^T, t \leftarrow t + 1$ 
8: end while
9: return  $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(t-1,p)}, \hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$ 
```

et al., 2015; Bertsekas, 2015). However, GS can be very inefficient in large-scale problems, as it needs to compute all partial derivatives. In fact, with the full gradient vector available, one could update the whole vector $\boldsymbol{\beta}$ at each step.

A recent breakthrough is the randomized coordinate gradient descent (Nesterov, 2012), which shows that random coordinate selection can achieve the same convergence rate as GS, and so is better than the cyclic rule (in an average sense). We use the random sampling without replacement (Wright, 2015). Concretely, at the start of each cycle, we randomly shuffle the elements in $[p]$ to obtain a new index set Perm[p], and then update the permuted coordinates sequentially. We present this **Accelerated-IPS (A-IPS)** with randomization in Algorithm 2.

4 A Majorization-Minimization Viewpoint

In this section, we use the majorization-minimization (MM) principle to study and generalize IPS. We refer to Lange et al. (2000) and Hunter and Lange (2004) for more details on MM algorithms. Rather than directly minimizing $l(\boldsymbol{\beta})$ in (5), our goal here is to construct a surrogate function $g(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-)$, with $\boldsymbol{\beta}^-$ denoting the value of $\boldsymbol{\beta}$ from the last iteration, such that the following properties are satisfied for all $\boldsymbol{\beta} \in \mathbb{R}^p$:

$$g(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) \geq l(\boldsymbol{\beta}), \quad g(\boldsymbol{\beta} \mid \boldsymbol{\beta}) = l(\boldsymbol{\beta}). \quad (11)$$

Then, if we set $\boldsymbol{\beta}^{(t+1)} \in \arg \min_{\boldsymbol{\beta}} g(\boldsymbol{\beta} \mid \boldsymbol{\beta}^{(t)})$, the sequence of iterates satisfies

$$l(\boldsymbol{\beta}^{(t+1)}) \leq g(\boldsymbol{\beta}^{(t+1)} \mid \boldsymbol{\beta}^{(t)}) \leq g(\boldsymbol{\beta}^{(t)} \mid \boldsymbol{\beta}^{(t)}) = l(\boldsymbol{\beta}^{(t)}).$$

That is, by minimizing a proper surrogate function $g(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-)$ iteratively, the objective function $l(\boldsymbol{\beta})$ is guaranteed to be monotonically non-increasing. There are other nice properties of the sequence of MM iterates since the problem under consideration is convex (see, e.g., Chapter 12 of [Lange \(2013\)](#)). We shall focus on the derivation of surrogate functions in this section. For notational simplicity, we define $\boldsymbol{\mu}^- \triangleq \exp(\mathbf{X}\boldsymbol{\beta}^-)$ and $\boldsymbol{\mu}^{(t)} \triangleq \exp(\mathbf{X}\boldsymbol{\beta}^{(t)})$. Once getting $\boldsymbol{\beta}^{(t+1)}$, we can update $\boldsymbol{\mu}$ according to

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} \circ \exp\left[\sum_j \mathbf{x}_j(\beta_j^{(t+1)} - \beta_j^{(t)})\right], \quad (12)$$

which results in an iterative proportional scaling.

4.1 GIS and extensions

We derive three surrogate functions all of which are applicable under the non-negative design setting (4). Recall the objective function:

$$l(\boldsymbol{\beta}) = \sum_i [q_i \exp(\tilde{\mathbf{x}}_i^T \boldsymbol{\beta}) - n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}] \triangleq \sum_i l_i. \quad (13)$$

Noticing the convexity of l_i in $\boldsymbol{\beta}$, we can apply Jensen's inequality

$$\begin{aligned} l_i &= -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + q_i \exp\left\{\sum_j \alpha_{ij} \left[\frac{x_{ij}}{\alpha_{ij}}(\beta_j - \beta_j^-)\right] + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^-\right\} \\ &\leq -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \sum_j \alpha_{ij} q_i \exp\left[\frac{x_{ij}}{\alpha_{ij}}(\beta_j - \beta_j^-) + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^-\right] \\ &= -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \sum_j \alpha_{ij} \mu_i^- \exp\left[\frac{x_{ij}}{\alpha_{ij}}(\beta_j - \beta_j^-)\right], \end{aligned} \quad (14)$$

where α_{ij} satisfy $\alpha_{ij} \geq 0$, $\sum_{j=1}^p \alpha_{ij} = 1$ for all i . We will not allow α_{ij} to be zero when $x_{ij} \neq 0$. If $x_{ij} = 0$, the associated term $x_{ij}\beta_j$ is β_j independent and so we set $\alpha_{ij} = 0$ for such x_{ij} formally. This gives

$$l_i \leq -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \sum_{\{j: x_{ij} \neq 0\}} \alpha_{ij} \mu_i^- \exp\left[\frac{x_{ij}}{\alpha_{ij}}(\beta_j - \beta_j^-)\right]. \quad (15)$$

Binary designs With the somewhat naive choice $\alpha_{ij} = 1/p$, we immediately obtain a surrogate satisfying (11):

$$g_1(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) = -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \frac{1}{p} \sum_{ij} \mu_i^- \exp[px_{ij}(\beta_j - \beta_j^-)]. \quad (16)$$

Given any binary design with $x_{ij} = 0$ or 1, which covers the contingency table setting, the optimal solution of $\arg \min_{\boldsymbol{\beta}} g_1(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-)$ is given by:

$$\beta_j = \beta_j^- + \frac{1}{p} \log[\mathbf{x}_j^T \mathbf{n} / (\mathbf{x}_j^T \boldsymbol{\mu}^-)], \quad j \in [p].$$

The updates of parameter and mean are obtained as follows

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \frac{1}{p} \log[(\mathbf{X}^T \mathbf{n}) \odot (\mathbf{X}^T \boldsymbol{\mu}^{(t)})] \quad (17)$$

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} \circ (\exp\{\frac{1}{p} \mathbf{X} \log[(\mathbf{X}^T \mathbf{n}) \odot (\mathbf{X}^T \boldsymbol{\mu}^{(t)})]\}). \quad (18)$$

The MM algorithm shares similarities with the IPS defined in Algorithm 1, but differs in two ways. IPS updates the components of $\boldsymbol{\beta}$ sequentially (asynchronizely), while (17) updates the entire vector synchronizely. Correspondingly, the stepsize in MM algorithm is smaller than that used in standard IPS.

Non-negative designs Recall (4), where we have $x_{i+} \neq 0$ for all i from assumption (iii) in Section 2. Setting $\alpha_{ij} = x_{ij}/x_{i+}$ in (15) yields

$$l_i \leq -\sum_j n_i x_{ij} \beta_j + \sum_j \frac{x_{ij}}{x_{i+}} \mu_i^- \exp[x_{i+}(\beta_j - \beta_j^-)], \quad (19)$$

which leads to another surrogate function

$$g_{20}(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) = -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \sum_{i,j} \frac{x_{ij}}{x_{i+}} \mu_i^- \exp[x_{i+}(\beta_j - \beta_j^-)].$$

The optimal $\boldsymbol{\beta}$ satisfies $\sum_i x_{ij} \mu_i^- \exp[x_{i+}(\beta_j - \beta_j^-)] = \sum_i n_i x_{ij}$ but does not enjoy a closed form solution in general.

We majorize the term $\exp[x_{i+}(\beta_j - \beta_j^-)]/x_{i+}$ further. For any a, b with $0 < a \leq b$ and $t \in \mathbb{R}$, $\exp(at)/a - 1/a \leq \exp(bt)/b - 1/b$. Applying the

inequality with $a = x_{i+}$ and $b = \max_i x_{i+} = \|\mathbf{X}\|_\infty \triangleq R$ ($R > 0$ from assumption (iii) in Section 2) to (19) gives

$$g_2(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) = -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \sum_{i,j} x_{ij} \mu_i^- \left\{ \frac{\exp[R(\beta_j - \beta_j^-)]}{R} - \frac{1}{R} + \frac{1}{x_{i+}} \right\}.$$

Now, the optimal solution can be explicitly evaluated: $\beta_j = \beta_j^- + (1/R) \log(\mathbf{x}_j^T \mathbf{n} / \mathbf{x}_j^T \boldsymbol{\mu}^-)$, and the iterates are then given by

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \frac{1}{R} \log[(\mathbf{X}^T \mathbf{n}) \oslash (\mathbf{X}^T \boldsymbol{\mu}^{(t)})] \quad (20)$$

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} \circ (\exp\{\frac{1}{R} \mathbf{X} \log[(\mathbf{X}^T \mathbf{n}) \oslash (\mathbf{X}^T \boldsymbol{\mu}^{(t)})]\}), \quad (21)$$

(20) is computationally more efficient than (17) in binary scenarios. (For example, for a three-way table of size $2 \times 2 \times 100$, the main-effects model has $p = 102$ and $R = 4$ and so the stepsize in (20) is much larger.)

This MM algorithm is the generalized iterative scaling proposed by Darroch and Ratcliff (1972) when $R = 1$. (Darroch and Ratcliff pre-transform the original design matrix such that all elements are non-negative and all row sums must be equal to one.)

General designs MM can handle a general design (4) without any transformation. We define two row and column support sets as $\mathcal{S}_i^c \triangleq \{j \in [p] \mid x_{ij} \neq 0\}$, $\mathcal{S}_j^r \triangleq \{i \in [N] \mid x_{ij} \neq 0\}$. Then

$$\begin{aligned} \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} &= \tilde{\mathbf{x}}_i^T (\boldsymbol{\beta} - \boldsymbol{\beta}^-) + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^- \\ &= \sum_{j \in \mathcal{S}_i^c} \frac{|x_{ij}|}{R} \left\{ \frac{R x_{ij} (\beta_j - \beta_j^-)}{|x_{ij}|} \right\} + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^- \\ &= \sum_{j \in \mathcal{S}_i^c} \frac{|x_{ij}|}{R} \left\{ \frac{R x_{ij} (\beta_j - \beta_j^-)}{|x_{ij}|} + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^- \right\} + (1 - \sum_j \frac{|x_{ij}|}{R}) \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^-, \end{aligned}$$

where $R = \|\mathbf{X}\|_\infty > 0$. Plugging it into l_i in (15) with $\alpha_{ij} = |x_{ij}|/R$, we obtain a new surrogate function

$$\begin{aligned} g_3(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) &= -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \sum_{i \in [N], j \in \mathcal{S}_i^c} \frac{\mu_i^- |x_{ij}|}{R} \exp\left[\frac{x_{ij}}{|x_{ij}|} R(\beta_j - \beta_j^-)\right] + \sum_i \mu_i^- (1 - \sum_j \frac{|x_{ij}|}{R}) \\ &= -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \sum_{j \in [p], i \in \mathcal{S}_j^r} \frac{\mu_i^- |x_{ij}|}{R} \exp\left[\frac{x_{ij}}{|x_{ij}|} R(\beta_j - \beta_j^-)\right] + \sum_i \mu_i^- (1 - \sum_j \frac{|x_{ij}|}{R}). \end{aligned}$$

Denote $x^p \triangleq (x + |x|)/2$ and $x^n \triangleq (-x + |x|)/2$ as x 's positive part and negative part, respectively, (both naturally extend to vectors). Then $\nabla_{\beta} g_3(\beta \mid \beta^-) = 0$ can be written as

$$-\mathbf{x}_j^T \mathbf{n} + \sum_i \mu_i^- x_{ij}^p \exp[R(\beta_j - \beta_j^-)] - \sum_i \mu_i^- x_{ij}^n \exp[-R(\beta_j - \beta_j^-)] = 0, \quad j \in [p],$$

where the left-hand side is quadratic in $\exp(R\beta_j)$. The update for β_j has an explicit form: $\beta_j = \beta_j^- + (1/R) \log\{(\mathbf{x}_j^T \mathbf{n} + \{(\mathbf{x}_j^p)^T \boldsymbol{\mu}^- + 4[(\mathbf{x}_j^p)^T \boldsymbol{\mu}^-][(\mathbf{x}_j^n)^T \boldsymbol{\mu}^-]\}^{\frac{1}{2}}) / [2(\mathbf{x}_j^p)^T \boldsymbol{\mu}^-]\}$ if $(\mathbf{x}_j^p)^T \boldsymbol{\mu}^- > 0$; $\beta_j = \beta_j^- + (1/R) \log\{[(\mathbf{x}_j^n)^T \boldsymbol{\mu}^-] / (-\mathbf{x}_j^T \mathbf{n})\}$ if $(\mathbf{x}_j^p)^T \boldsymbol{\mu}^- = 0$, $(\mathbf{x}_j^n)^T \boldsymbol{\mu}^- \neq 0$ or $\mathbf{x}_j^T \mathbf{n} \neq 0$; $\beta_j = \beta_j^-$ otherwise.

In the special case of a non-negative design, $(\mathbf{x}_j^p)^T \boldsymbol{\mu}^- > 0$ and $\mathbf{x}_j^n = \mathbf{0}$. Thus above update degenerates to (20) and it extends the MM update (20) to an arbitrary design matrix.

4.2 A reparametrization trick and IIS

In this subsection, we assume that \mathbf{X} contains a column corresponding to the intercept, i.e.,

$$\mathbf{X} = [\mathbf{1} \ \dot{\mathbf{X}}] \in \mathbb{R}^{N \times p} \quad \text{and} \quad \beta = [\beta_0 \ \dot{\beta}^T]^T \in \bar{\mathbb{R}}^p, \quad (22)$$

where the intercept β_0 is a scalar and $\dot{\beta} \in \bar{\mathbb{R}}^{p-1}$ denotes the slope vector. Introducing $\alpha \triangleq \beta_0 + \log\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle$, or $\exp(\alpha) = \langle \mathbf{q}, \exp(\beta_0 \mathbf{1} + \dot{\mathbf{X}} \dot{\beta}) \rangle = \langle \mathbf{1}, \boldsymbol{\mu} \rangle$, $l(\beta)$ in (5) becomes

$$\begin{aligned} l(\beta) &= -\langle \mathbf{n}, \beta_0 \mathbf{1} + \dot{\mathbf{X}} \dot{\beta} \rangle + \langle \mathbf{q}, \exp(\beta_0 \mathbf{1} + \dot{\mathbf{X}} \dot{\beta}) \rangle \\ &= -\langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle - \langle \mathbf{1}, \mathbf{n} \rangle \beta_0 + \exp \alpha \\ &= -\langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle - \langle \mathbf{1}, \mathbf{n} \rangle [\alpha - \log\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle] + \exp(\alpha). \end{aligned}$$

The minimization problem with respect to α can be solved by $\exp(\alpha) = \langle \mathbf{1}, \mathbf{n} \rangle = \langle \mathbf{1}, \boldsymbol{\mu} \rangle$, which means the optimal β_0 satisfies

$$\beta_0 = \log\langle \mathbf{1}, \mathbf{n} \rangle - \log\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle. \quad (23)$$

Therefore, it suffices to study the $\dot{\beta}$ -optimization:

$$\min_{\dot{\beta} \in \bar{\mathbb{R}}^{p-1}} L(\dot{\beta}) \triangleq -\langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle + \langle \mathbf{1}, \mathbf{n} \rangle \log\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle. \quad (24)$$

An MM algorithm can be developed for (24). Specifically, due to the concavity of the log function, for any $\zeta > 0$, we have $\log(\zeta x) \leq \zeta x - 1$ and this bound is tight with choice $\zeta(x) = 1/x$. Then, we get

$$\begin{aligned} L(\dot{\beta}) &= -\langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle + \langle \mathbf{1}, \mathbf{n} \rangle \log[\zeta \langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle] - \langle \mathbf{1}, \mathbf{n} \rangle \log \zeta \\ &\leq -\langle \mathbf{1}, \mathbf{n} \rangle \log \zeta - \langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle + \langle \mathbf{1}, \mathbf{n} \rangle [\zeta \langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}) \rangle - 1]. \end{aligned} \quad (25)$$

Assuming the non-negative setting (4) and applying Jensen's inequality (15) with $\alpha_{ij} = \dot{x}_{ij}/\dot{x}_{i+}$, we immediately obtain

$$\begin{aligned} g_3(\dot{\beta} \mid \dot{\beta}^-) &= -\langle \mathbf{n}, \dot{\mathbf{X}} \dot{\beta} \rangle - \langle \mathbf{1}, \mathbf{n} \rangle (\log \zeta + 1) \\ &\quad + \zeta \langle \mathbf{1}, \mathbf{n} \rangle \sum_{i \in [N], j \in [p-1]} \frac{\dot{x}_{ij}}{\dot{x}_{i+}} \dot{\mu}_i^- \exp[\dot{x}_{i+}(\dot{\beta}_j - \dot{\beta}_j^-)], \end{aligned} \quad (26)$$

where $\dot{\mu} \triangleq \mathbf{q} \circ \exp(\dot{\mathbf{X}} \dot{\beta})$ with its past value denoted by $\dot{\mu}^- = \mathbf{q} \circ \exp(\dot{\mathbf{X}} \dot{\beta}^-)$. The only choice to guarantee (11) is $\zeta = 1/\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\beta}^-) \rangle$.

Now, minimizing g_3 in (26), $\dot{\beta}_j^{(t+1)}$ is determined by:

$$\frac{\langle \mathbf{1}, \mathbf{n} \rangle}{\langle \mathbf{1}, \dot{\mu}^{(t)} \rangle} \sum_i \dot{x}_{ij} \dot{\mu}_i^{(t)} \exp[\dot{x}_{i+}(\dot{\beta}_j^{(t+1)} - \dot{\beta}_j^{(t)})] = \sum_i n_i \dot{x}_{ij}, \quad j \in [p-1], \quad (27)$$

and the proportional scaling of $\dot{\mu}$ is according to:

$$\dot{\mu}^{(t+1)} = \dot{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})]. \quad (28)$$

Interestingly, this algorithm has a close connection to the celebrated IIS (Pietra et al., 1997) that is based on the *normalized* observed and estimated counts, i.e., $\bar{\mathbf{n}} \triangleq \mathbf{n}/\langle \mathbf{1}, \mathbf{n} \rangle$ and $\bar{\mu} \triangleq \mu/\langle \mathbf{1}, \mu \rangle$.

Theorem 3. *The sequence of $\{\dot{\beta}^{(t)}\}_{t=0}^\infty$ generated from the MM algorithm (27) and (28) coincides with the $\{\dot{\beta}^{(t)}\}_{t=0}^\infty$ generated by IIS:*

$$\sum_i \bar{n}_i \dot{x}_{ij} = \sum_i \dot{x}_{ij} \bar{\mu}_i^{(t)} \exp[\dot{x}_{i+}(\dot{\beta}_j^{(t+1)} - \dot{\beta}_j^{(t)})], \quad j \in [p-1] \quad (29)$$

$$\bar{\mu}^{(t+1)} = \bar{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})]/\langle \bar{\mu}^{(t)}, \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})] \rangle. \quad (30)$$

(27) and (28) do not need the normalization operation nor the intercept update during the iteration. One can extract the intercept estimate based on (23) at the end.

4.3 Quadratic surrogates and accelerations

The reparametrized form (24) offers more options in constructing surrogate functions. A pleasing fact is that the term $\langle \mathbf{1}, \mathbf{n} \rangle \log \langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\boldsymbol{\beta}}) \rangle$ has uniformly bounded curvature. In the following, we focus on a quadratic surrogate function Q , given a general design satisfying (22)

$$Q(\dot{\boldsymbol{\beta}} \mid \dot{\boldsymbol{\beta}}^-) = L(\dot{\boldsymbol{\beta}}^-) + \langle \dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\beta}}^-, \nabla_{\dot{\boldsymbol{\beta}}} L(\dot{\boldsymbol{\beta}}^-) \rangle + \frac{1}{2} (\dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\beta}}^-)^T \mathbf{W} (\dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\beta}}^-). \quad (31)$$

By Taylor expansion, Q is a valid surrogate function provided that

$$\langle \mathbf{1}, \mathbf{n} \rangle \dot{\mathbf{X}}^T \left[\frac{\text{diag}(\dot{\boldsymbol{\mu}})}{\langle \mathbf{1}, \dot{\boldsymbol{\mu}} \rangle} - \frac{\dot{\boldsymbol{\mu}} \dot{\boldsymbol{\mu}}^T}{\langle \mathbf{1}, \dot{\boldsymbol{\mu}} \rangle^2} \right] \dot{\mathbf{X}} \preceq \mathbf{W}, \quad \forall \dot{\boldsymbol{\mu}} \succeq \mathbf{0}. \quad (32)$$

A straightforward choice is

$$\mathbf{W} = (\langle \mathbf{1}, \mathbf{n} \rangle \|\dot{\mathbf{X}}\|_2^2 / 2) \mathbf{I}. \quad (33)$$

A finer one is due to [Bohning and Lindsay \(1988\)](#)

$$\mathbf{W} = \dot{\mathbf{X}}^T (\langle \mathbf{1}, \mathbf{n} \rangle \mathbf{I} - \mathbf{1} \mathbf{1}^T) \dot{\mathbf{X}} / 2. \quad (34)$$

Based on our experience, (34) is better in most large data problems and this choice will be adopted unless otherwise specified. In either case, the optimal solution of $\arg \min_{\dot{\boldsymbol{\beta}}} Q(\dot{\boldsymbol{\beta}} \mid \dot{\boldsymbol{\beta}}^-)$ can be written as $\dot{\boldsymbol{\beta}}^- - \mathbf{W}^{-1} [-\dot{\mathbf{X}}^T \mathbf{n} + (\langle \mathbf{1}, \mathbf{n} \rangle / \langle \mathbf{1}, \dot{\boldsymbol{\mu}}^- \rangle) \dot{\mathbf{X}}^T \dot{\boldsymbol{\mu}}^-]$.

Since the last term $(\dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\beta}}^-)^T \mathbf{W} (\dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\beta}}^-) / 2$ in (31) can be identified as a Bregman divergence $\mathbf{D}(\dot{\boldsymbol{\beta}} \mid \dot{\boldsymbol{\beta}}^-)$, (31) falls into the computational framework studied by [Nesterov \(1988\)](#) and the so-called Nesterov's second acceleration scheme applies. The resulting accelerated algorithm is provided in Algorithm 3, referred to as the **Quadratic-surrogate IPS (Q-IPS)**.

5 Further Extensions and Variants

5.1 Block-wise IPS on big data

The principles of CD and MM can be applied more flexibly to reduce a large difficult problem into a series of smaller and easier sub-problems (which may

Algorithm 3 Q-IPS.

Input \mathbf{n} , \mathbf{q} , $\dot{\mathbf{X}}$, and \mathbf{W} from (34)

Initialize $\dot{\boldsymbol{\beta}}^{(0)} \in \mathbb{R}^{p-1}$, $\theta_0 \leftarrow 1$, $t \leftarrow 0$

1: $\dot{\boldsymbol{\mu}}^{(0)} \leftarrow \mathbf{q} \circ \exp(\dot{\mathbf{X}} \dot{\boldsymbol{\beta}}^{(0)})$, $\dot{\boldsymbol{\eta}}^{(0)} \leftarrow \dot{\boldsymbol{\beta}}^{(0)}$

2: **while** not converged **do**

3: $\dot{\boldsymbol{\alpha}}^{(t)} \leftarrow (1 - \theta_t) \dot{\boldsymbol{\beta}}^{(t)} + \theta_t \dot{\boldsymbol{\eta}}^{(t)}$

4: $\dot{\boldsymbol{\eta}}^{(t+1)} \leftarrow \arg \min_{\dot{\boldsymbol{\beta}}} [\langle \nabla_{\dot{\boldsymbol{\beta}}} L(\dot{\boldsymbol{\alpha}}^{(t)}), (\dot{\boldsymbol{\beta}} - \dot{\boldsymbol{\alpha}}^{(t)}) \rangle + \theta_t \mathbf{D}(\dot{\boldsymbol{\beta}} \mid \dot{\boldsymbol{\eta}}^{(t)})]$, i.e.,

$$\dot{\boldsymbol{\eta}}^{(t+1)} \leftarrow \dot{\boldsymbol{\eta}}^{(t)} - \theta_t^{-1} \mathbf{W}^{-1} \left\{ -\dot{\mathbf{X}}^T \mathbf{n} + \frac{\langle \mathbf{1}, \mathbf{n} \rangle}{\langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\boldsymbol{\alpha}}^{(t)}) \rangle} \dot{\mathbf{X}}^T [\mathbf{q} \circ \exp(\dot{\mathbf{X}} \dot{\boldsymbol{\alpha}}^{(t)})] \right\}$$

5: $\dot{\boldsymbol{\beta}}^{(t+1)} \leftarrow (1 - \theta_t) \dot{\boldsymbol{\beta}}^{(t)} + \theta_t \dot{\boldsymbol{\eta}}^{(t+1)}$

6: $\dot{\boldsymbol{\mu}}^{(t+1)} \leftarrow \dot{\boldsymbol{\mu}}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\boldsymbol{\beta}}^{(t+1)} - \dot{\boldsymbol{\beta}}^{(t)})]$

7: $\theta_{t+1} \leftarrow (\sqrt{\theta_t^4 + 4\theta_t^2} - \theta_t^2)/2$

8: $t \leftarrow t + 1$

9: **end while**

10: $\beta_0^{(t)} \leftarrow \log \langle \mathbf{1}, \mathbf{n} \rangle - \log \langle \mathbf{q}, \exp(\dot{\mathbf{X}} \dot{\boldsymbol{\beta}}^{(t)}) \rangle$

11: $\boldsymbol{\mu}^{(t)} \leftarrow \exp(\beta_0^{(t)}) \dot{\boldsymbol{\mu}}^{(t)}$

12: **return** $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(t)}$, $\hat{\boldsymbol{\beta}} = [\beta_0^{(t)} \ \dot{\boldsymbol{\beta}}^{(t)T}]^T$

not be univariate) so that Newton-type methods can be applied with ease. Two illustrations are given as follows.

We first try the block-wise coordinate descent (BCD, as an extension of CD) on the reparametrized problem (24). Let $\{G_1, \dots, G_k, \dots, G_m\}$ form a partition of set $[p-1]$ with $|G_k| = g_k$ for $1 \leq k \leq m$. We write $\dot{\boldsymbol{\beta}} = [\dot{\boldsymbol{\beta}}_1^T \dots \dot{\boldsymbol{\beta}}_k^T \dots \dot{\boldsymbol{\beta}}_m^T]^T$ with $\dot{\boldsymbol{\beta}}_k$ the sub-vector of $\dot{\boldsymbol{\beta}}$ indexed by G_k , and $\dot{\mathbf{X}}_k$ the sub-matrix of $\dot{\mathbf{X}}$ associated with $\dot{\boldsymbol{\beta}}_k$. Fixing the other blocks of coefficients at $\dot{\boldsymbol{\beta}}_j^-$ ($j \neq k$), the sub-problem of $\dot{\boldsymbol{\beta}}_k$ is $\min_{\dot{\boldsymbol{\beta}}_k} L(\dot{\boldsymbol{\beta}}_1^-, \dots, \dot{\boldsymbol{\beta}}_{k-1}^-, \dot{\boldsymbol{\beta}}_k, \dot{\boldsymbol{\beta}}_{k+1}^-, \dots, \dot{\boldsymbol{\beta}}_m^-)$. Since g_k can be made sufficiently small, Newton's method can be applied based on the following gradient and Hessian

$$\nabla_{\dot{\boldsymbol{\beta}}_k} L(\dot{\boldsymbol{\beta}}_k) = -\dot{\mathbf{X}}_k^T \mathbf{n} + \frac{\langle \mathbf{1}, \mathbf{n} \rangle}{\langle \mathbf{1}, \dot{\boldsymbol{\mu}} \rangle} \dot{\mathbf{X}}_k^T \dot{\boldsymbol{\mu}}, \quad \nabla_{\dot{\boldsymbol{\beta}}_k}^2 L(\dot{\boldsymbol{\beta}}_k) = \langle \mathbf{1}, \mathbf{n} \rangle \dot{\mathbf{X}}_k^T \left[\frac{\text{diag}(\dot{\boldsymbol{\mu}})}{\langle \mathbf{1}, \dot{\boldsymbol{\mu}} \rangle} - \frac{\dot{\boldsymbol{\mu}} \dot{\boldsymbol{\mu}}^T}{\langle \mathbf{1}, \dot{\boldsymbol{\mu}} \rangle^2} \right] \dot{\mathbf{X}}_k.$$

Quasi-Newton approaches such as L-BFGS can be used as well. Two crucial

factors determining the performance are (i) the rule of selecting a proper block to update at each iteration, and (ii) the block sizes g_k .

Similar to the discussions in Section 3.3, the GS rule requires much fewer iteration steps than the plain cyclic rule. In terms of the total computational time, however, GS incurs significantly more overhead in big data applications, due to its calculation of the full gradient in block selection. Perhaps surprisingly, some well-known random rules such as block resampling without replacement do not always have satisfactory performance, either (cf. B-IPS (FB) in Fig 1). We recommend random blocking followed by cyclic update. Specifically, at the beginning of each iteration, all coordinates are randomly shuffled and then sequentially assigned into m blocks with sizes g_1, \dots, g_m . Next, we update the blocks of coefficients in a cyclic manner. This beats other update rules in our experiments. The scalable randomized block-wise IPS (**B-IPS**), presented in Algorithm 4, combines the virtues of IPS and Newton-type methods.

Algorithm 4 B-IPS.

Input $\mathbf{n}, \mathbf{q}, \mathring{\mathbf{X}}$ and block sizes g_1, g_2, \dots, g_m

Initialize $\mathring{\boldsymbol{\beta}}^{(0)} \in \mathbb{R}^{p-1}, t \leftarrow 0$

```

1: while not converged do
2:   Sequentially assign the elements in set Perm[ $p-1$ ] to  $m$  blocks with
   sizes  $g_1, g_2, \dots, g_m$ 
3:    $\mathring{\boldsymbol{\mu}}^{(t,0)} \leftarrow \mathbf{q} \circ \exp(\mathring{\mathbf{X}} \mathring{\boldsymbol{\beta}}^{(t)})$ 
4:   for  $k = 1, 2, \dots, m$  do
5:      $\mathring{\boldsymbol{\beta}}_k^{(t+1)} \in \arg \min_{\mathring{\boldsymbol{\beta}}_k} L(\mathring{\boldsymbol{\beta}}_1^{(t+1)}, \dots, \mathring{\boldsymbol{\beta}}_{k-1}^{(t+1)}, \mathring{\boldsymbol{\beta}}_k, \mathring{\boldsymbol{\beta}}_{k+1}^{(t)}, \dots, \mathring{\boldsymbol{\beta}}_m^{(t)})$ 
6:      $\mathring{\boldsymbol{\mu}}^{(t,k)} \leftarrow \mathring{\boldsymbol{\mu}}^{(t,k-1)} \circ \exp[\mathring{\mathbf{X}}_k(\mathring{\boldsymbol{\beta}}_k^{(t+1)} - \mathring{\boldsymbol{\beta}}_k^{(t)})]$ 
7:   end for
8:    $\mathring{\boldsymbol{\beta}}^{(t+1)} \leftarrow [\mathring{\boldsymbol{\beta}}_1^{(t+1)T}, \dots, \mathring{\boldsymbol{\beta}}_m^{(t+1)T}]^T, t \leftarrow t + 1$ 
9: end while
10:  $\beta_0^{(t)} \leftarrow \log \langle \mathbf{1}, \mathbf{n} \rangle - \log \langle \mathbf{q}, \exp(\mathring{\mathbf{X}} \mathring{\boldsymbol{\beta}}^{(t)}) \rangle$ 
11:  $\boldsymbol{\mu}^{(t)} \leftarrow \exp(\beta_0^{(t)}) \mathring{\boldsymbol{\mu}}^{(t-1,m)}$ 
12: return  $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(t)}, \hat{\boldsymbol{\beta}} = [\beta_0^{(t)} \mathring{\boldsymbol{\beta}}^{(t)T}]^T$ 

```

The block sizes play an important role in speeding the computation. Suppose all g_k 's are roughly equal, then as one increases g_k , the total number of blocks decreases. This may cause Newton's method to be less efficient on

each sub-problem, but is helpful in serial computation since the number of such optimization problems drops. In experience, having g_k between 100 and 400 often achieves a good balance when using Newton as the sub-problem solver.

Different from the sequential update of B-IPS, MM is capable of deriving algorithms that update all blocks in parallel. Consider (13) with non-negative designs. Let $\{G_1, \dots, G_k, \dots, G_m\}$ form a partition of the whole set $[p]$ and $\tilde{\mathbf{x}}_{i,k}^T$ be the i th row vector of the sub-matrix \mathbf{X}_k . Similar to the derivations of (14)–(15), we have

$$\begin{aligned} l_i &= -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + q_i \exp\left\{\sum_k \alpha_{ik} \left[\frac{1}{\alpha_{ik}} \tilde{\mathbf{x}}_{i,k}^T (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^-)\right] + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^-\right\} \\ &\leq -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \sum_k \alpha_{ik} q_i \exp\left[\frac{1}{\alpha_{ik}} \tilde{\mathbf{x}}_{i,k}^T (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^-) + \tilde{\mathbf{x}}_i^T \boldsymbol{\beta}^-\right] \\ &= -n_i \tilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \sum_{\{k: \tilde{\mathbf{x}}_{i,k} \neq \mathbf{0}\}} \alpha_{ik} \mu_i^- \exp\left[\frac{1}{\alpha_{ik}} \tilde{\mathbf{x}}_{i,k}^T (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^-)\right], \end{aligned}$$

where α_{ik} satisfy $\alpha_{ik} \geq 0$, $\sum_{k=1}^m \alpha_{ik} = 1$ for all i . Letting $x_{i+} = \langle \mathbf{1}, \tilde{\mathbf{x}}_i \rangle$, $x_{i+,k} = \langle \mathbf{1}, \tilde{\mathbf{x}}_{i,k} \rangle$ and $\alpha_{ik} = x_{i+,k}/x_{i+}$, we have a surrogate as follows

$$g_4(\boldsymbol{\beta} \mid \boldsymbol{\beta}^-) = -\mathbf{n}^T \mathbf{X} \boldsymbol{\beta} + \sum_{i,k} \frac{x_{i+,k}}{x_{i+}} \mu_i^- \exp\left[\frac{x_{i+}}{x_{i+,k}} \tilde{\mathbf{x}}_{i,k}^T (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^-)\right].$$

If all G_k 's are singletons, g_4 degenerates to g_{20} . A nice feature is that g_4 is separable in $\boldsymbol{\beta}_k$ and so all blocks of coefficients can be simultaneously updated. Although the convergence of the MM algorithm is typically slower than that of BCD, with parallel computing resources available, we anticipate it to have potential advantages in big data applications.

5.2 Shrinkage estimation

Modern statistical applications often involve a large number of variables, where regularization is necessary to achieve better estimation accuracy or model parsimony. For example, one can append an ℓ_2 -type penalty to the negative log-likelihood to handle collinearity:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f_2(\boldsymbol{\beta}) \triangleq -\langle \mathbf{n}, \mathbf{X} \boldsymbol{\beta} \rangle + \langle \mathbf{q}, \exp(\mathbf{X} \boldsymbol{\beta}) \rangle + \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2, \quad (35)$$

where $\lambda > 0$ is a regularization parameter and can be tuned by, say, AIC (Akaike, 1974). Empirically, even setting λ at a small value (e.g. 1e-6) often improves estimation accuracy. We recommend including a mild ℓ_2 -penalty in all practical applications, especially those with zero counts. (35) does not add much difficulty in algorithm design. Most of the previously developed algorithms such as IIS, Q-IPS, B-IPS can be easily modified to adapt to (35) and the details are omitted here.

Another popular regularization is through the ℓ_1 -penalty. This can help practitioners select among various association models. Many aforementioned optimization techniques still apply. In the following, we use CD to derive a fast ℓ_1 -algorithm for contingency tables. Assuming a binary design \mathbf{X} (cf. (4)), we consider the problem

$$\min_{\beta \in \mathbb{R}^p} f_1(\beta) \triangleq -\langle \mathbf{n}, \mathbf{X}\beta \rangle + \langle \mathbf{q}, \exp(\mathbf{X}\beta) \rangle + \lambda \|\beta\|_1, \quad (36)$$

where $\lambda > 0$. In this subsection, we denote $\mathbf{X} = [\mathbf{x}_0 \dots \mathbf{x}_{p-1}]$ for notational convenience. To derive the coordinate-wise update, it suffices to study a univariate version of (36).

Lemma 1. *Let $\mathbf{n}, \mathbf{x} \in \mathbb{R}^N$ and $\beta \in \mathbb{R}$. Then the solution to the optimization problem $\min_{\beta \in \mathbb{R}} -\langle \mathbf{n}, \mathbf{x} \rangle \beta + \langle \mathbf{q}, \exp(\beta \mathbf{x}) \rangle + \lambda |\beta|$ is given by $\hat{\beta} = \log\{[\langle \mathbf{x}, \mathbf{n} \rangle - \lambda \operatorname{sgn}(\langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle)] / \langle \mathbf{x}, \mathbf{q} \rangle\}$ if $|\langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle| \geq \lambda$ and $\hat{\beta} = 0$ otherwise.*

Let $\boldsymbol{\mu}^{(t,j-1)} = \mathbf{q} \circ \exp(\mathbf{X}[\beta_0^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j^{(t)}, \dots, \beta_{p-1}^{(t)}]^T)$. From the lemma, $\min_{\beta_j \in \mathbb{R}} f_1(\beta_0^{(t+1)}, \dots, \beta_{j-1}^{(t+1)}, \beta_j, \beta_{j+1}^{(t)}, \dots, \beta_{p-1}^{(t)})$ gives

$$\beta_j^{(t+1)} = \begin{cases} \beta_j^{(t)} + \log\left[\frac{\langle \mathbf{x}_j, \mathbf{n} \rangle - \lambda \operatorname{sgn}(\delta_j)}{\langle \mathbf{x}_j, \boldsymbol{\mu}^{(t,j-1)} \rangle}\right] & |\delta_j| \geq \lambda \\ 0 & |\delta_j| < \lambda, \end{cases} \quad (37)$$

where $1 \leq j \leq p-1$, $\delta_j \triangleq \langle \mathbf{x}_j, \mathbf{n} - \boldsymbol{\mu}^{(t,j-1)} \circ \exp(-\beta_j^{(t)} \mathbf{x}_j) \rangle = \langle \mathbf{x}_j, \mathbf{n} - \boldsymbol{\mu}^{(t,j-1)} \exp(-\beta_j^{(t)}) \rangle$. The intercept is not subject to the penalty and is updated according to $\beta_0^{(t+1)} = \beta_0^{(t)} + \log(\langle \mathbf{x}_0, \mathbf{n} \rangle / \langle \mathbf{x}_0, \boldsymbol{\mu}^{(t,-1)} \rangle)$. Overall, one just needs to replace Step 4 of Algorithm 1 by (37). This modified algorithm is denoted by ℓ_1 -IPS in experiments. Note that ℓ_1 -IPS does not need any quadratic approximation or line search. Algorithms similar to B-IPS can be developed based on Schmidt (2010), which will be investigated in future work.

6 Experiments

6.1 Setup

We test different algorithms on log-linear models with offset $\mathbf{q} = \mathbf{1}$, where contingency tables as well as non-negative and general designs (cf. (4)) are included. We assume that the intercept exists, i.e. $\mathbf{X} = [\mathbf{1} \ \mathring{\mathbf{X}}]$. Given \mathbf{X} and β^* , the mean is $\mu^* = \mathbf{q} \circ \exp(\mathbf{X}\beta^*)$, and observed counts $\mathbf{n} = [n_i]$ can be independently generated by $n_i \sim \text{Poi}(\mu_i)$. Given each setting, we generate 20 simulation data sets and report averaged results based on the relative gradient $\|\mathbf{g}_t\|_\infty / \|\mathbf{g}_0\|_\infty$ and the relative estimation error $\|\beta^{(t)} - \beta^*\|_2^2 / \|\beta^*\|_2^2$, where \mathbf{g}_t ($t \geq 0$) is the gradient of the original objective function (5) evaluated at $\beta^{(t)}$. These two measures help characterize the computational performance and the statistical performance of a given algorithm. The termination criterion is made based on the size of relative gradient

$$\|\mathbf{g}_t\|_\infty / \|\mathbf{g}_0\|_\infty \leq \epsilon_{\text{tol}}, \quad (38)$$

and the running time limit (e.g., $t_{\text{max}} \leq 600$). By default, $\epsilon_{\text{tol}} = 1\text{e-}4$. Computational error and statistical error will be plotted against computational time.

The algorithms for comparison are listed in Table 1. We set $\beta^{(0)} = \mathbf{0}$ in all experiments. The block sizes in B-IPS are set to be $g_k = 200$ unless otherwise specified. For Newton’s method, we adopt the `minFunc` package with bracketing line-search steps (Schmidt, 2012). Algorithms implementations are in Matlab R2016a and the simulation experiments are performed on an Intel Core I5-4460S machine with 16GB RAM.

Algorithms	Description
IPS	Algorithm 1
A-IPS	Algorithm 2
GIS	(20) and (21)
IIS	(29) and (30)
Q-IPS	Algorithm 3
B-IPS	Algorithm 4
Newton	(Schmidt, 2012)

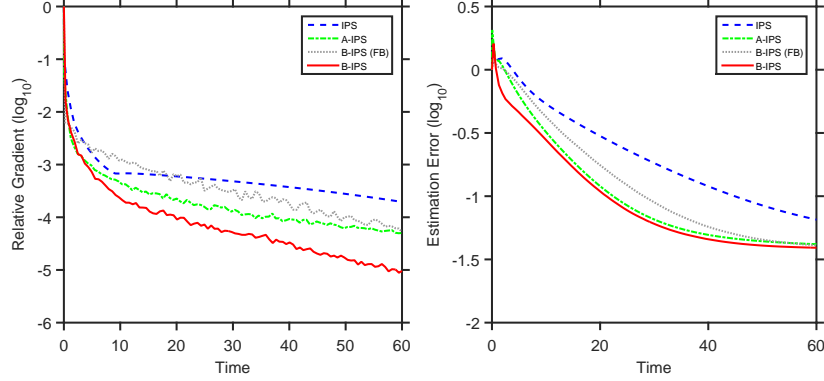
Table 1: Algorithms in simulation experiments.

6.2 Contingency tables: power of randomization

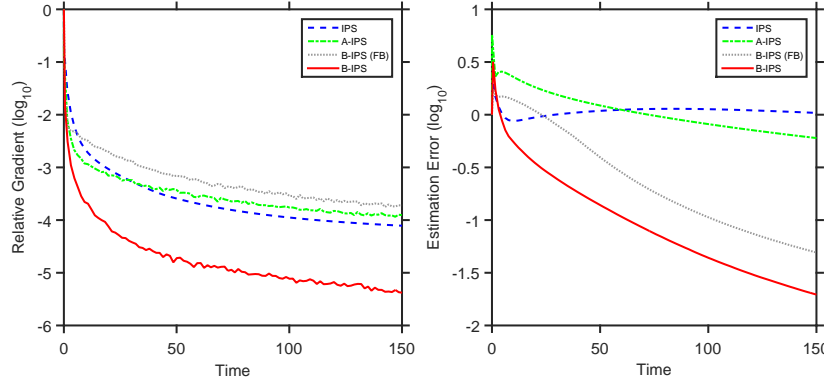
Moderate-sized tables We consider moderate-sized tables with 4 categorical variables each taking 10 distinct levels, namely, a table of size $10 \times 10 \times 10 \times 10$. We assume a two-way homogeneous association model, which includes all effects up to two-way interactions (Agresti, 2012). This corresponds to a design with one intercept column, 36 main effects and 486 two way associations ($p = 523$).

We generate coefficients in two different settings: (i) all components of β^* are non-zero except the last 10 coordinates, which are sampled independently according to $0.5\mathcal{N}(1, 1) + 0.5\mathcal{N}(3, 1)$, and (ii) in addition to the last 10 coordinates, another 20 randomly chosen coordinates are set to be non-zero following the same distribution. The intercept is set to be $\beta_0^* = 2$.

The results averaged over 20 runs are shown in Fig 1. In computation, A-IPS is much faster than IPS in setting (i), and is comparable to it in setting (ii). In both scenarios, A-IPS delivers more accurate estimates. Fixed blocking with random selection, denoted by B-IPS (FB), does not demonstrate the full power of block-wise computation. B-IPS with random blocking is clearly the winner both in optimization and in statistical estimation.



(a) Relative gradient (left) and estimation error (right) (both on a \log scale) in setting (i).



(b) Relative gradient (left) and estimation error (right) (both on a \log scale) in setting (ii).

Figure 1: Performance comparison between IPS, A-IPS, and B-IPS (with FB denoting “fixed blocking”) in moderate-sized table settings ($N = 10,000$, $p = 523$).

Large tables In this experiment, we demonstrate the scalability of A-IPS and B-IPS on even larger tables. We use 5 categorical variables each taking 10 distinct levels to construct a table of size $10 \times 10 \times 10 \times 10 \times 10$. We consider a three-way homogeneous association model including all interactions up to third order, which gives $p = 8,146$. β^* is generated such that $\beta_0^* = 5$ and the last 2,000 coordinates independently follow $\mathcal{N}(1, 1)$ with the rest as zero.

We compare Newton’s method, A-IPS and B-IPS in Fig 2 (IPS is too slow and thus dropped). According to the figure, Newton’s algorithm does

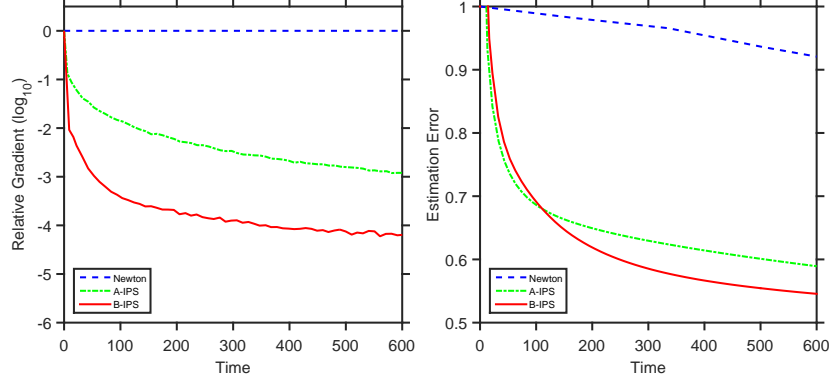


Figure 2: Relative gradient and estimation error of Newton, A-IPS, and B-IPS in a large table setting ($N = 100,000$, $p = 8,146$) on the scale of \log_{10} .

not deliver a useful estimate within the time limit, and we notice that it is quite memory-demanding in the example. In comparison, A-IPS and B-IPS offer great scalability and statistical accuracy.

6.3 Log-linear models: generalizations and scalability

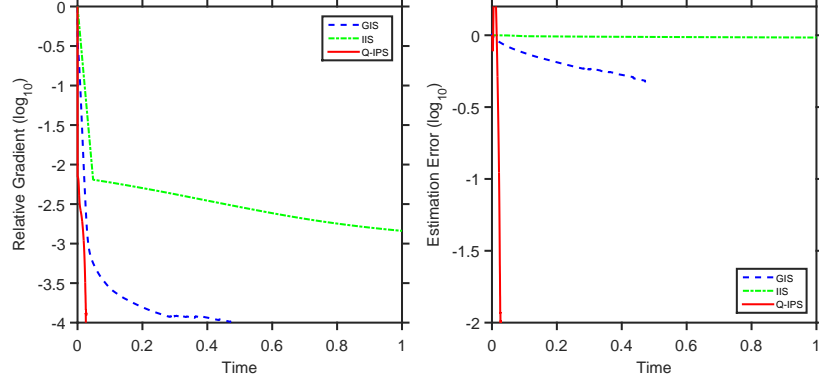
In this subsection, we apply some extensions of IPS to fit Poisson log-linear models. The prototype design $\mathring{\mathbf{X}}$ (excluding the intercept) is generated with each row sampled independently from $\mathcal{N}(\mathbf{0}, [\rho^{|j-k|}])$, where $\rho = 0.8$ and $1 \leq j, k \leq p-1$. Some shifting and scaling operations will be performed to make the design matrix non-negative or to avoid overflow issues.

6.3.1 Non-negative designs

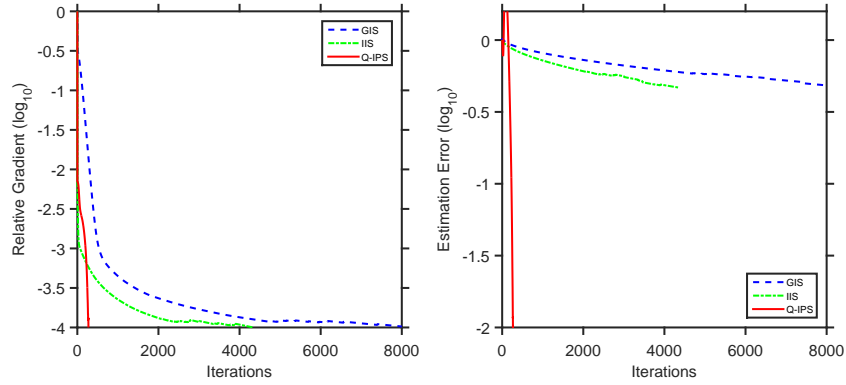
To apply GIS or IIS, a non-negative design matrix is required. This can be done by $\mathring{\mathbf{X}} \leftarrow \mathring{\mathbf{X}} - (\min_{i,j} \mathring{x}_{ij}) \mathbf{1}\mathbf{1}^T$. We scale it down further to achieve better numerical stability $\mathring{\mathbf{X}} \leftarrow \mathring{\mathbf{X}} / (50 \|\mathring{\mathbf{X}}\|_{\max})$. The obtained matrix has all row sums roughly equal (an artifact of the iid design), which appears too ideal in real-world data. So we multiply each row by a random factor $1 + |z_i|$ with $z_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. The true coefficient vector is generated as $\beta_0^* = 1$ and $\beta_j^* \stackrel{\text{i.i.d.}}{\sim} 0.5\mathcal{N}(10, 1) + 0.5\mathcal{N}(-10, 1)$ for $1 \leq j \leq p-1$.

A relatively small-scale experiment with $N = 1,000$ and $p = 100$ is per-

formed to find out the winning algorithm among GIS, IIS and Q-IPS. The results are shown in Fig 3, when we plot the algorithm progress in terms of computational time (3a) and iteration number (3b). (To better differentiate GIS and Q-IPS, we only show part of the plot.) IIS is much more time-consuming than GIS and Q-IPS. A careful examination shows that the per-iteration cost of IIS is pretty high, as one has to solve $p - 1$ nonlinear equations. On the other hand, IIS converges in fewer number of iterations than GIS. We anticipate IIS to show better efficiency with faster equation solvers or with parallel computing. In both scenarios, Q-IPS outperforms GIS and IIS.



(a) Relative gradient and estimation error (shown on a **log** scale) as a function of time.



(b) Relative gradient and estimation error (shown on a **log** scale) with respect to the iteration numbers.

Figure 3: Performance comparison between GIS, IIS and Q-IPS in a small non-negative setting ($N = 1,000$, $p = 100$).

We then conduct a larger experiment to compare Q-IPS, B-IPS and Newton. We set $N = 20,000$ with $p = 2,000$ or $4,000$. (GIS and IIS are excluded as they are too slow.) In generating the simulation data, we again scale down the prototype design by $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} / (20 \|\hat{\mathbf{X}}\|_{\max})$, and set $\beta_j^* \stackrel{\text{i.i.d}}{\sim} 0.5\mathcal{N}(10, 1) + 0.5\mathcal{N}(-10, 1)$ for $1 \leq j \leq p - 1$ and $\beta_0^* = 10$. See Fig 4 for the results.

The first row of Fig 4 demonstrates Newton-Raphson's two typical stages: the damped Newton phase and the quadratically convergent phase (Boyd and Vandenberghe, 2004). Though converging super fast in the second stage, this

method takes too long to complete the first stage when $p = 4,000$, thereby less useful in big data applications. In contrast, B-IPS and Q-IPS are able to deliver meaningful estimates within the given time limit. B-IPS seems to have better scalability than Q-IPS in both situations.

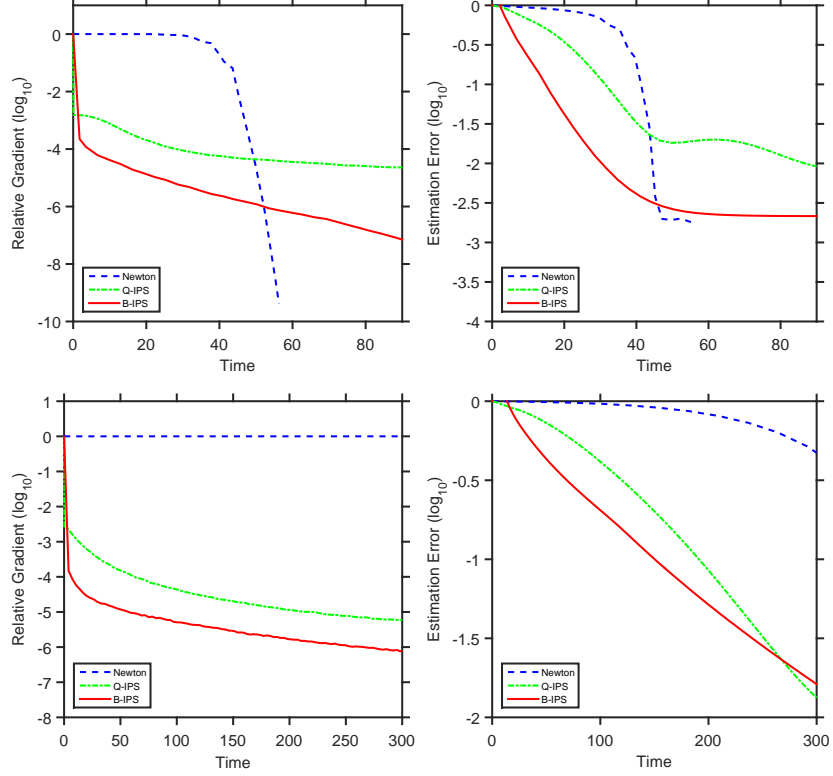


Figure 4: Performances comparison between Q-IPS, B-IPS and Newton on large non-negative designs with $N = 20,000$, $p = 2,000$ (top) and 4,000 (bottom).

6.3.2 General designs

In this part, we compare the performance of Newton, B-IPS and L-BFGS on general designs. We set $\beta_0^* = 10$ and $\beta_j^* \stackrel{\text{i.i.d}}{\sim} 0.5\mathcal{N}(10, 1) + 0.5\mathcal{N}(-10, 1)$ for $1 \leq j \leq p - 1$ with the prototype design matrix scaled down by $\mathring{\mathbf{X}} \leftarrow \mathbf{X} / (100\|\mathbf{X}\|_{\max})$.

Fixing $N = 50,000$, we vary p from 1,000 to 12,000 and report the results in Table 2. When $p = 10,000$ or 12,000, since not all methods converge fast, we set a time limit $t_{\max} = 600$. The relative gradient (relGrad) and the estimation error (estErr) are scaled by $1e+7$ and $1e+4$, respectively.

Table 2: Optimization and statistical performances of Newton, B-IPS and L-BFGS on general designs of large size.

	$p = 1,000$ $\epsilon_{\text{tol}} = 1e-6$		$p = 4,000$ $\epsilon_{\text{tol}} = 1e-6$		$p = 10,000$ $t_{\max} = 600$	
	Time	estErr	Time	estErr	relGrad	estErr
Newton	35.2	0.12	528.7	0.11	–	–
B-IPS ¹	16.8	3.2	116.4	23.2	5.92	109.3
	$p = 4,000$ $\epsilon_{\text{tol}} = 1e-6$		$p = 10,000$ $t_{\max} = 600$		$p = 12,000$ $t_{\max} = 600$	
	Time	estErr	relGrad	estErr	relGrad	estErr
L-BFGS	276.5	2.6	12250	1022	–	–
B-IPS ²	94.2	2.8	5.6	44.1	12.8	462.8

According to Table 2, Newton’s method, when feasible, gives the smallest estimation error. But it easily fails when p is large (say $p \geq 6000$). B-IPS¹ (with Newton as the sub-problem solver and $g_k = 200$) has better efficiency and scalability in computation, which can be observed from $p = 10,000$. The same conclusion can be drawn from the comparison between quasi-Newton and B-IPS. In our experiments, L-BFGS (limited-memory BFGS, [Liu and Nocedal \(1989\)](#)) is much less expensive than Newton, but still runs out of memory when $p > 10,000$. B-IPS², which takes L-BFGS as the sub-problem solver and sets $g_k = 2000$, has the best scalability. Overall, the benefits of B-IPS, brought by BCD, randomization and reparametrization, are impressive in large-scale problems. We also noticed that B-IPS is less precise than Newton in general, but having low computational complexity makes it suitable for big data applications where moderate accuracy usually suffices.

6.4 ℓ_1 -IPS on real data

The dataset is collected from a Portuguese marketing campaign related to bank deposit subscription ([Moro et al., 2014](#)). We use 41,188 instances and

10 categorical variables to study whether a client subscribes a term deposit or not. The information of these variables is shown in Table 3. We group the data at each observed combination level of all categorical variables, and use the total number of successful subscriptions as the response variable.

Variables	Description
job	type of job (12 levels)
marital	marital status (4 levels)
education	education level (8 levels)
default	whether a client has credit in default (3 levels)
housing	whether a client has housing loan (3 levels)
loan	whether a client has personal loan (3 levels)
contact	contact communication type, cellular or telephone (2 levels)
month	the month in which the last contact was made (10 levels)
day	the day of week when the last contact was made (5 levels)
poutcome	outcome of the previous marketing campaign (3 levels)

Table 3: Categorical variables and their descriptions.

We consider a three-way association model, including all main effects, second-order interactions, and third-order interactions. This results in a large model with 5,874 predictors. We ran ℓ_1 -IPS to compute the solution path as shown in Fig 5 and used EBIC (Chen and Chen, 2008) for parameter tuning. A sparse model with 16 predictors is obtained, see Table 4. (We also performed the same experiment on a two-way association model and found that all the main terms and two-way interaction terms in Table 4 still got selected.)

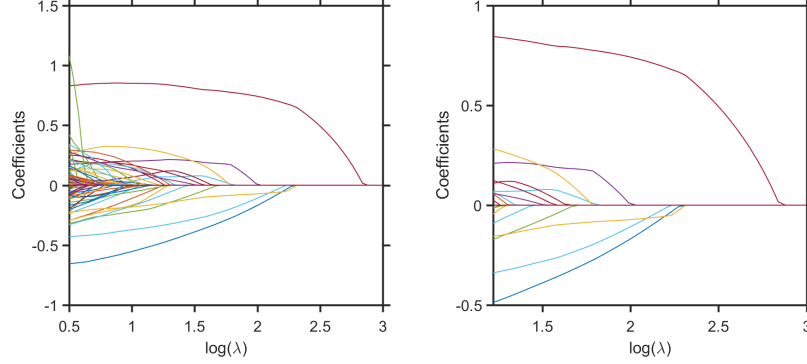


Figure 5: Left: the ℓ_1 solution path on the bank marketing data. Right: the solution path including selected variables only.

Some meaningful conclusions can be drawn from the variable selection result. First, as supported by previous studies (e.g., [Moro et al. \(2014\)](#)), `month` plays an important role in the marketing campaign. In particular, clients are unlikely to subscribe a term deposit if the last contact occurs in November. Moreover, contact communication type being cellular and the outcome of previous marketing campaign being successful are favorable factors for successful subscription.

According to the table, all of the selected two-way interactions involve `poutcome = non-exist` (no record in the previous campaign), which separates out this special group of clients. Our model also contains two three-way terms, and it is worth mentioning that they show no collinearity with the other selected variables. This indicates that the group of married technicians who received professional training and the group of singles with university-level degrees but no record in the previous marketing campaign tend to subscribe a term deposit.

Main	month = mar (+)	month = may (-)
	month = oct (+)	month = nov (-)
	education = univ (+)	poutcome = success (+)
	default = unkown (-)	contact = telephone (-)
	loan = yes (-)	
Two-way	(poutcome = non-exist) * month = mar (+)	
	(poutcome = non-exist) * month = oct (+)	
	(poutcome = non-exist) * education = univ (+)	
	(poutcome = non-exist) * contact = telephone (-)	
	(poutcome = non-exist) * housing = unkown (-)	
Three-way	(job = technician) * (education = professional)	
	* (marital = married) (+)	
	(poutcome = non-exist) * education = univ	
	* (marital = single) (+)	

Table 4: Selected variables with signs of coefficients in parentheses.

A Appendix

A.1 Proof of Theorem 2

The result is an application of Theorem 2.1 of [Luo and Tseng \(1992\)](#). In fact, under the assumptions made in Section 2 and the assumption in this theorem, $g(\mathbf{X}\beta) \triangleq \mathbf{q}^T \exp(\mathbf{X}\beta)$ is strictly convex and twice continuously differentiable on its effective domain, and $\nabla^2 g(\mathbf{X}\hat{\beta})$ is positive definite. It follows from Theorem 2.1 of [Luo and Tseng \(1992\)](#) that $\{\beta^{(t)}\}_{t=0}^\infty$ generated by the Algorithm 1 (with cyclic update) converges to $\hat{\beta}$ at least linearly.

A.2 Proof of Theorem 3

We prove by induction. Given $\hat{\beta}^{(t)}$, $\hat{\mu}^{(t)}$, $\bar{\mu}^{(t)}$ with $\bar{\mu}^{(t)} = \hat{\mu}^{(t)} / \langle \mathbf{1}, \hat{\mu}^{(t)} \rangle$, it suffices to show that (i) equation (27) and equation (29) yield the same solution, and (ii) $\bar{\mu}^{(t+1)}$ obtained from (30) and $\hat{\mu}^{(t+1)}$ updated from (28) satisfy $\bar{\mu}^{(t+1)} = \hat{\mu}^{(t+1)} / \langle \mathbf{1}, \hat{\mu}^{(t+1)} \rangle$.

Plugging $\bar{\mu}^{(t)} = \hat{\mu}^{(t)} / \langle \mathbf{1}, \hat{\mu}^{(t)} \rangle$ into (27) gives:

$$\langle \mathbf{1}, \mathbf{n} \rangle \sum_i \hat{x}_{ij} \bar{\mu}_i^{(t)} \exp[\hat{x}_{i+} (\hat{\beta}_j^{(t+1)} - \hat{\beta}_j^{(t)})] = \sum_i n_i \hat{x}_{ij},$$

or equivalently

$$\sum_i \dot{x}_{ij} \bar{\mu}_i^{(t)} \exp[\dot{x}_{i+}(\dot{\beta}_j^{(t+1)} - \dot{\beta}_j^{(t)})] = \sum_i (n_i / \langle \mathbf{1}, \mathbf{n} \rangle) \dot{x}_{ij},$$

which is exactly (29).

Next, given $\dot{\beta}^{(t+1)}$, we have

$$\begin{aligned} \frac{\dot{\mu}^{(t+1)}}{\langle \mathbf{1}, \dot{\mu}^{(t+1)} \rangle} &= \frac{\dot{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})]}{\langle \mathbf{1}, \dot{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})] \rangle} \\ &= \frac{[\dot{\mu}^{(t)} / \langle \mathbf{1}, \dot{\mu}^{(t)} \rangle] \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})]}{\langle \mathbf{1}, [\dot{\mu}^{(t)} / \langle \mathbf{1}, \dot{\mu}^{(t)} \rangle] \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})] \rangle} \\ &= \frac{\bar{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})]}{\langle \mathbf{1}, \bar{\mu}^{(t)} \circ \exp[\dot{\mathbf{X}}(\dot{\beta}^{(t+1)} - \dot{\beta}^{(t)})] \rangle} \\ &= \bar{\mu}^{(t+1)}, \end{aligned}$$

where the first equality is due to (27), the third follows from the induction hypothesis, and the last is due to (30).

A.3 Proof of Lemma 1

Let $f_0(\beta) = -\langle \mathbf{n}, \mathbf{x} \rangle \beta + \langle \mathbf{q}, \exp(\beta \mathbf{x}) \rangle + \lambda |\beta|$. From the Karush–Kuhn–Tucker equation

$$\mathbf{x}^T \boldsymbol{\mu} - \mathbf{x}^T \mathbf{n} + \lambda s(\beta) = 0, \quad (39)$$

where $\boldsymbol{\mu} = \mathbf{q} \circ \exp(\beta \mathbf{x})$ and the sub-gradient $s(\beta)$ satisfies $s(\beta) = \text{sgn}(\beta)$ if $\beta \neq 0$ and $s(\beta) \in [-1, 1]$ if $\beta = 0$. For such a convex problem, this equation is necessary and sufficient for the solution.

Recall (8), and so we have $\mathbf{x}^T \boldsymbol{\mu} = \langle \mathbf{x}, \mathbf{q} \rangle \exp \beta$. Plugging this into (39) gives $\langle \mathbf{x}, \mathbf{q} \rangle \exp \beta - \mathbf{x}^T \mathbf{n} + \lambda s(\beta) = 0$. The solution thus follows

$$\hat{\beta} = \begin{cases} \log \left(\frac{\langle \mathbf{x}, \mathbf{n} \rangle - \lambda \langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle}{\langle \mathbf{x}, \mathbf{q} \rangle} \right) & \langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle \geq \lambda \\ 0 & -\lambda < \langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle < \lambda \\ \log \left(\frac{\langle \mathbf{x}, \mathbf{n} \rangle + \lambda \langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle}{\langle \mathbf{x}, \mathbf{q} \rangle} \right) & \langle \mathbf{x}, \mathbf{n} - \mathbf{q} \rangle \leq -\lambda. \end{cases}$$

References

- Agresti, A. (2012). *Categorical data analysis*. Wiley, New York, third edition.
- Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.
- Beck, A. and Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060.
- Bertsekas, Dimitri P, A. (2015). *Convex Optimization Algorithms*. Athena Scientific, Belmont, MA.
- Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1975). *Discrete multivariate analysis: theory and practice*. The MIT Press, Cambridge, MA.
- Bohning, D. and Lindsay, B. G. (1988). Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663.
- Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press, Cambridge.
- Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.
- Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pages 146–158.
- Darroch, J. N. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, pages 1470–1480.
- Deming, W. E. and Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444.
- Fienberg, S. E. (1970). An iterative procedure for estimation in contingency tables. *The Annals of Mathematical Statistics*, pages 907–917.
- Fienberg, S. E. and Meyer, M. M. (2006). Iterative proportional fitting. *Encyclopedia of Statistical Sciences*, 6:3723–3726.

- Fienberg, S. E. and Rinaldo, A. (2012). Maximum likelihood estimation in log-linear models. *Ann. Statist.*, 40(2):996–1023.
- Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332.
- Good, I. J. (1963). Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables. *The Annals of Mathematical Statistics*, pages 911–934.
- Haberman, S. J. (1974). *The analysis of frequency data*. The University of Chicago Press, Chicago. Statistical Research Monographs, Vol. IV.
- Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37.
- Ireland, C. T. and Kullback, S. (1968). Contingency tables with given marginals. *Biometrika*, 55(1):179–188.
- Kurras, S. (2015). Symmetric iterative proportional fitting. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 526–534.
- Lahr, M. and De Mesnard, L. (2004). Biproportional techniques in input-output analysis: table updating and structural analysis. *Economic Systems Research*, 16(2):115–134.
- Lange, K. (2013). *Optimization*. Springer, New York, second edition.
- Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20.
- Lauritzen, S. L. (1996). *Graphical models*, volume 17 of *Oxford Statistical Science Series*. The Clarendon Press, Oxford University Press, New York. Oxford Science Publications.
- Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

- Luo, Z.-Q. and Tseng, P. (1992). On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35.
- Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.
- Nesterov, Y. (1988). On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Matem Metody*, 24(3):509–517.
- Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362.
- Nutini, J., Schmidt, M., Laradji, I. H., Friedlander, M., and Koepke, H. (2015). Coordinate descent converges faster with the gauss-southwell rule than random selection. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1632–1641.
- Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):380–393.
- Pukelsheim, F. (2014). Biproportional scaling of matrices and the iterative proportional fitting procedure. *Annals of Operations Research*, 215(1):269–283.
- Razaviyayn, M., Hong, M., and Luo, Z.-Q. (2013). A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153.
- Rote, G. and Zachariasen, M. (2007). Matrix scaling by network flow. In *Symposium on Discrete Algorithms: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, volume 7, pages 848–854.
- Schmidt, M. (2010). *Graphical model structure learning with l_1 -regularization*. PhD thesis.
- Schmidt, M. (2012). minfunc: unconstrained differentiable multivariate optimization in matlab.

- Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879.
- Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.
- Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.
- Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.